

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 INTRODUCTION

In the current generation, we find it difficult to remember the names of relatives further than one generation. Every time we see our far away relatives, we are not able to recognize who that person is and what their relationship is with us. We forget the birthday and wedding anniversary of relatives who are far away from us, both by distance and generation. We forget their address and phone number; it is always difficult to find the contact number on our phone if there is someone else with the same name. Our Family Database Management System will resolve all these problems and provide many more features.

This report describes the development and implementation of a web-based Family Management System for managing and inviting family members to events. The system was built using the Django web framework and utilizes a relational database to store and organize information about family members, couples, families, events, and invitations. The system allows for the creation, editing, and deletion of events, as well as the invitation of individual family members, couples, or entire families to events. Additionally, the system also allows for storing detailed personal information of family members, including contact information, occupation, and parent/children details.[1]

The main objective of this system is to provide a user-friendly, efficient, and secure way of managing family member information and event invitations. The system is designed to be easily extensible and customizable, making it suitable for various use cases. The system was developed to meet the needs of various organizations and individuals who are looking for an easy way to manage and invite family members to events. The system is designed to keep the users updated with their family members and events they are invited to. The system is also designed to keep the data secure and organized to avoid any errors and confusion. The purpose of this system is to enable the user to manage their family members, events and personal information related to them in an organized and efficient way.[1]

## 1.2 SCOPE OF THE PROJECT

The scope of this project includes the development and implementation of a web-based system for managing and inviting family members to events. The system is designed to provide the following features:

1. Creation, editing, and deletion of members.
2. Creation, editing and deletion of events.
3. Invitation of individual family members, couples, or entire families to events
4. Storing detailed personal information of family members, including contact information, occupation, and parent/children's details.
5. User-friendly and efficient interface for managing and organizing family member information and event invitations.
6. Security measures to protect the stored data such as log in and sign up to access the website.
7. Customizable and extensible design to suit various use cases.[1]

This project will be focused on the development of the system and its core features. The system is designed to be easily extensible to meet the needs of various organizations and individuals who are looking for an easy way to manage and invite family members to events. The system will be deployed and tested with real-world data.

## 1.3 FEATURES OF FAMILY MANAGEMENT SYSTEM

Here, is the list of all the features currently available/implemented on our system:

1. Event Management: Allows for the creation, and deletion of events, as well as the ability to view a list of upcoming events.
2. Family Member Management: Allows for the creation, editing, and deletion of family member information, as well as the ability to view a list of all family members.

3. Couple Management: Allows for the creation of couple information with their family member IDs.
4. Family Management: Allows for the creation of family information, as well as the ability to view a list of all families.
5. Personal Information Management: Allows for the creation, and editing of personal information of a family member like contact details and occupation.
6. Parent/Children Management: Allows for the creation of parent/children information of a family member.
7. Event Invitation: Allows for the invitation of individual family members, couples, or entire families to events.
8. User-friendly interface: A user-friendly and efficient interface for managing and organizing family member information and event invitations.[1]
9. Reports: The system can generate reports like invitees list, family member list, couple list, etc.
10. User Management: The system can create, edit, and delete users and assign them different roles and permissions. This is an Admin only accessible feature.
11. User authentication and Authorization: The system has a secure authentication and authorization mechanism to ensure that only authorized users can access the system.
12. Activity Logs: The system can keep track of user activities and generate activity logs. This is an Admin only accessible feature.[1]

## **1.4 IMPORTANCE OF FAMILY MANAGEMENT SYSTEM**

The proposed system is important for several reasons. Firstly, it provides a user-friendly and efficient way of managing family member information and event invitations, making it easy for users to keep track of their family members and events they are invited to. This can save time and effort for the user, especially for large families or organizations with

many events. Secondly, the system provides security measures to protect the stored data, ensuring that the user's personal information is kept confidential and protected from unauthorized access. Thirdly, the system is designed to be easily extensible, making it suitable for various use cases. This means that the system can be customized to meet the specific needs of different organizations and individuals.

Finally, the system will also be helpful for organizations and individuals who are looking for an easy way to manage and invite family members to events. The system will help them to keep track of their events, family members and personal information.

In summary, the proposed system is important as it provides a user-friendly and efficient way to manage family member information and event invitations, ensures the security of the stored data, and is easily extensible to meet various use cases. It will be helpful for organizations and individuals who are looking for an easy way to manage and invite family members to events.[1]

## **CHAPTER 2**

# **SOFTWARE REQUIREMENT SPECIFICATION**

## REQUIREMENT ANALYSIS

The Requirements of a system are the descriptions of the services that a system should provide and the constraints on its operation. These requirements reflect the need of customers for a system that serves a certain purpose such as controlling a device, or finding information.

The Family Database Management System has its requirements specification too. We can classify our requirements as User and System.

### 2.1 USER REQUIREMENTS

User requirements are the statements in natural language and explains what are the services expected from the system. The users of our system will be anyone who wishes to store their family members data onto the database and retrieve their personal data and relationship with other family members when requested. Keeping that in mind we can specify the user requirements as:

1. The system must be able to store the data of every family member.
2. If we enter the ID of a family member and search for their parents, the system must be able to tell the user who is the parent of that member.
3. If we enter the couple's details and search for their children, the system must be able to tell the user who the children are.
4. When we enter the ID of a member and request for personal information, it must display the details.
5. When we enter the details of an event and the invitees for it, the system must be able to display the details of all the invitees for that event.
6. The system must be able to keep a track of the count of the family members in each family.

We can explain the requirements in much more detail with the system specifications and requirements.

## **2.2 SYSTEM REQUIREMENTS SPECIFICATION**

System or Software requirements are much more detailed specifications of user requirements but with a software developers' point of view. It will give a deeper understanding of what the system is supposed to do and are classified as functional and non-functional requirements. The functional and non-functional requirements of our system are specified as follows

### **2.2.1 FUNCTIONAL REQUIREMENTS**

Functional requirements are the specific actions or tasks that a system must be able to perform. These requirements define what the system must do and are usually expressed in terms of inputs, processing, and outputs.

The functional requirements of Family Database Management System are:

1. Create and delete event details.
2. Invite individual family members, couples, or entire families to events.
3. Create, edit, and delete family member information.
4. Create, and delete couple information.
5. Create, and delete family information.
6. Create, edit, and delete personal information of a family member. The personal info will be phone number, address, email, and Occupation.
7. Create, and delete parent/children information of a family member.
8. View a list of upcoming events.
9. View a list of all family members.
10. Generate reports like invitees list, family member list, couple list, etc.



11. Create, edit, and delete users and assign them different roles and permissions.  
(Admin accessible feature)[1]
12. Secure authentication and authorization mechanism.[1]
13. Keep track of user activities and generate activity logs. [1]

### **2.2.2 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements are the requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and memory use. They may also define constraints on the system implementation, such as the capabilities of I/O devices or the data representations used in interfaces with other systems. They are much more critical.

The Non-Functional requirements of Family Database Management System are:

1. The System must be deployed as a web page available to all users.
2. It must be a secure website; in technical terms it must be https. This is mainly due to the system's feature to store the personal details of family members and these details cannot be disclosed to unauthorized users.
3. It must have a good response time. The system must return the relationship value within a few seconds.
4. The entire UI of the webpage must be smooth for the users.
5. The server that would store the database must have sufficient storage.
6. User-friendly and efficient interface
7. Security measures to protect the stored data
8. Customizable and extensible design.[1]
9. High performance and scalability
10. High availability and reliability
11. Compliance with relevant regulations and standards

12. Accessibility and compatibility with different devices and browsers
13. Easy to use and understand user manual and technical documentation.

## **2.3 SOFTWARE AND HARDWARE SPECIFICATIONS**

### **2.3.1 SOFTWARE REQUIREMENTS:**

1. Backend: Django 4.1.3, Python 3.10

A short note on Django: The entire system backend is built on a Django framework. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. It follows the model-view-controller (MVC) architectural pattern and emphasizes reusability and "pluggability" of components. Some of its key features include an automatic admin interface, a robust Object-Relational Mapping (ORM) system for database access, and support for template-based rendering of pages. Django also includes built-in support for user authentication, form handling, and other common web development tasks. It is an open-source framework and widely used for building web applications.[1]

2. Additional Packages: mysql-connector-python 8.0.31, mysqlclient 2.1.1
3. Frontend: HTML5, CSS
4. Database Language: MySQL
5. IDE: VS Code
6. Repository Server: GitHub
7. UML Diagram Design Software: Lucidchart
8. Browser: Brave 1.47, Chrome 109, Edge 109

### **2.3.2 HARDWARE REQUIREMENTS:**

1. Processor: AMD Ryzen-7 5800H, intel i5-8th gen and above
2. RAM: 2GB or more

3. Secondary Storage for database: 2 GB or more
4. Monitor: 1024x768 screen resolution
5. Keyboard and mouse

## **CHAPTER 3**

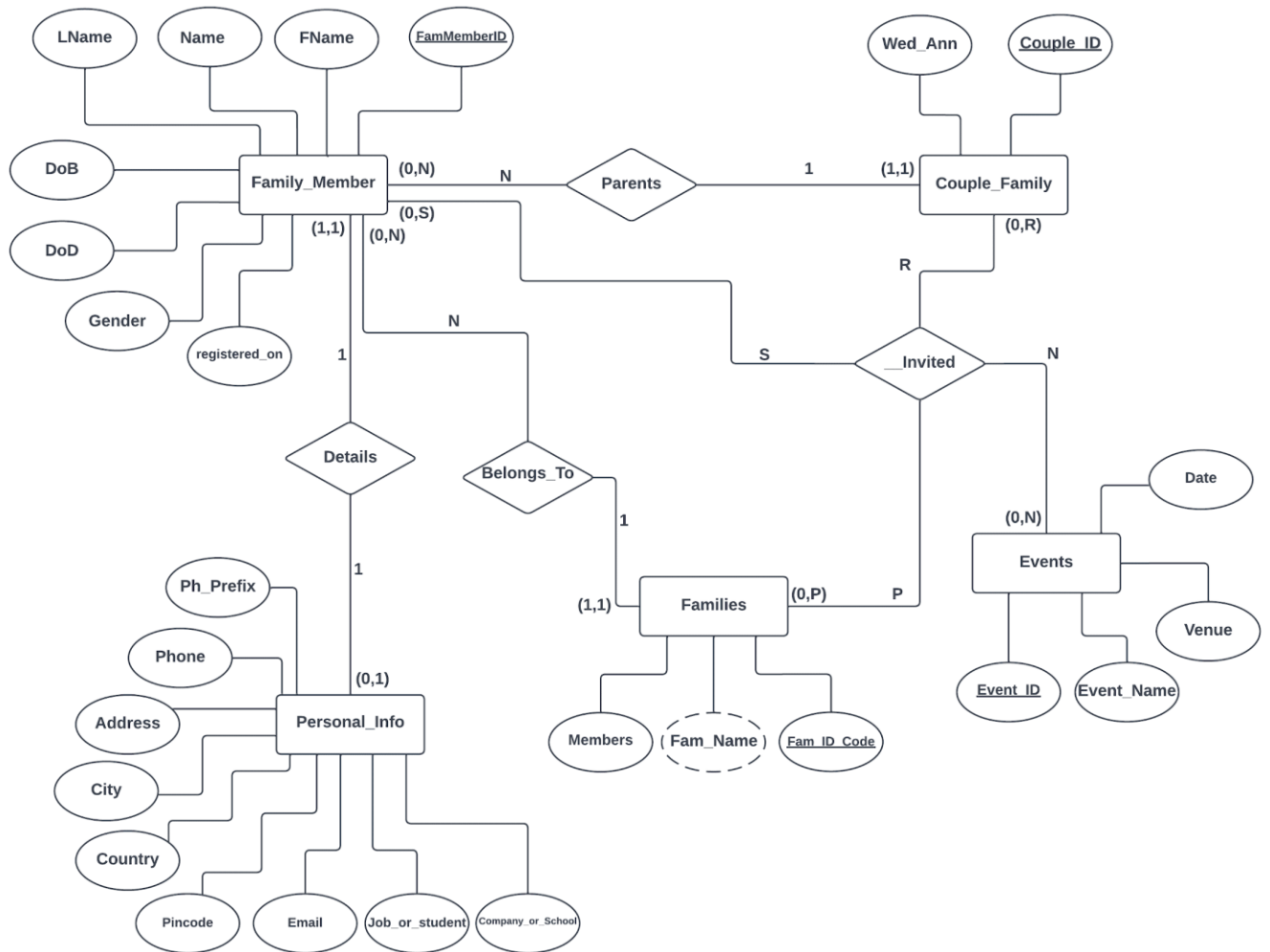
# **SYSTEM DESIGN**

## 3.1 DATABASE DESIGN

### 3.1.1 Entities, attributes, and relationship:

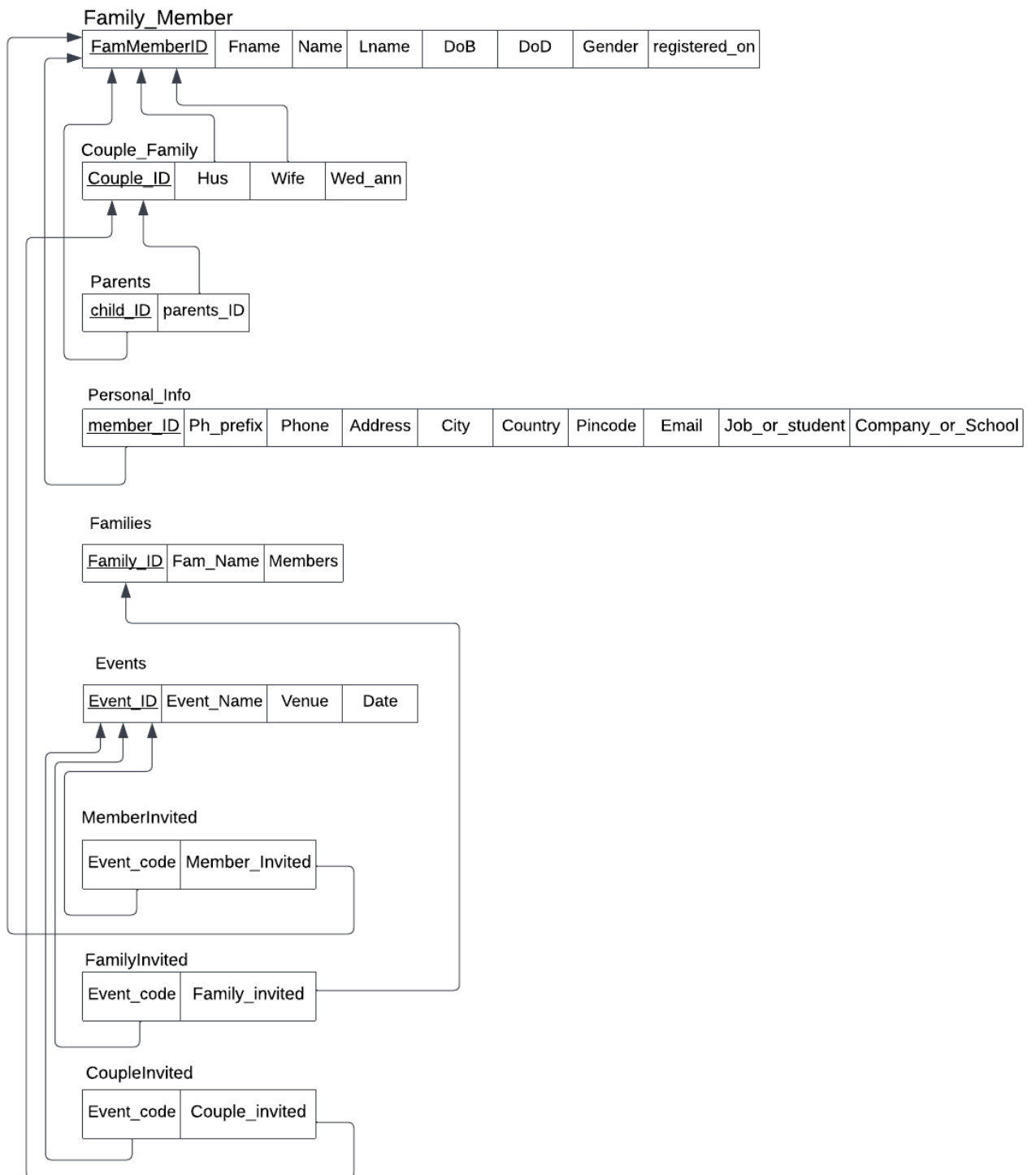
1. Family\_Member ( FamMemberID, FName, Name, LName, DoB, DoD, Gender, registered\_on )
2. Couple\_Family ( Couple\_ID, Hus\_ID, Wife\_ID, Wed\_Ann )
3. Parents ( ID, child\_ID\_id, parents\_ID\_id )
4. Personal\_Info ( member\_ID\_id, Ph\_Prefix, Phone, Address, City, Country, Pincode, Email, Job\_or\_student, Company\_or\_School )
5. Families ( Family\_ID, Fam\_Name, Members )
6. Events ( Event\_ID, Event\_Name, Venue, Date )
7. Member\_invited ( id, Event\_code\_id, Member\_Invited\_id )
8. Family\_invited ( id, Event\_code\_id, Family\_Invited\_id )
9. Couple\_invited ( id, Event\_code\_id, Couple\_Invited\_id )

### 3.2 ER DIAGRAM



F.3.2

### 3.3 SCHEMA DIAGRAM



F.3.3

### 3.4 Table Descriptions:

#### Family\_Member:

**T.3.4.1**

Field	Type	Null	Key	Default
FamMemberID	VARCHAR(20)	NO	PRI	NULL
FName	VARCHAR(30)	NO		NULL
Name	VARCHAR(30)	NO		NULL
LName	VARCHAR(30)	NO		NULL
DoB	DATE	NO		NULL
DoD	DATE	YES		NULL
Gender	VARCHAR(2)	NO		NULL
registered_on	DATETIME(6)	NO		NULL

#### Couple\_Family:

**T.3.4.2**

Field	Type	Null	Key	Default
Couple_ID	VARCHAR(20)	NO	PRI	NULL
Wed_Ann	DATE	NO		NULL
Hus_ID	VARCHAR(20)	NO	MUL	NULL
Wife_ID	VARCHAR(20)	NO	MUL	NULL

#### Parents:

**T.3.4.3**

Field	Type	Null	Key	Default	Extra
ID	INT	NO	PRI	NULL	AUTO_INCREMENT
child_ID_id	VARCHAR(20)	NO	MUL	NULL	
parents_ID_id	VARCHAR(20)	NO	MUL	NULL	



**Personal\_Info:****T.3.4.4**

Field	Type	Null	Key	Default
member_ID_id	VARCHAR(20)	NO	PRI	NULL
Ph_Prefix	VARCHAR(6)	NO		NULL
Phone	VARCHAR(18)	NO		NULL
Address	VARCHAR(100)	NO		NULL
City	VARCHAR(30)	NO		NULL
Country	VARCHAR(30)	NO		NULL
Pincode	VARCHAR(15)	YES		NULL
Email	VARCHAR(254)	NO		NULL
Job_or_student	VARCHAR(50)	NO		NULL
Company_or_School	VARCHAR(50)	NO		NULL

**Families:****T.3.4.5**

Field	Type	Null	Key	Default
Family_ID	VARCHAR(20)	NO	PRI	NULL
Fam_Name	VARCHAR(50)	NO		NULL
Members	INT	NO		NULL

**Events:****T.3.4.6**

Field	Type	Null	Key	Default	Extra
Event_ID	INT	NO	PRI	NULL	AUTO_INCREMENT
Event_Name	VARCHAR(100)	NO		NULL	
Venue	VARCHAR(100)	NO		NULL	
Date	DATE	NO		NULL	

**Member\_invited:****T.3.4.7**

Field	Type	Null	Key	Default	Extra
id	BIGINT	NO	PRI	NULL	AUTO_INCREMENT
Event_code_id	INT	NO	MUL	NULL	
Member_Invited_id	VARCHAR(20)	YES	MUL	NULL	

**Family\_invited:****T.3.4.8**

Field	Type	Null	Key	Default	Extra
id	BIGINT	NO	PRI	NULL	AUTO_INCREMENT
Event_code_id	INT	NO	MUL	NULL	
Family_Invited_id	VARCHAR(20)	YES	MUL	NULL	

**Couple\_invited:****T.3.4.9**

Field	Type	Null	Key	Default	Extra
id	BIGINT	NO	PRI	NULL	AUTO_INCREMENT
Event_code_id	INT	NO	MUL	NULL	
Couple_Invited_id	VARCHAR(20)	YES	MUL	NULL	

## **CHAPTER 4**

## **SCREENSHOTS**

## 4.1 THE WEBSITE

### 4.1.1 Log-in and Sign-Up [2][6]



**LOG IN**

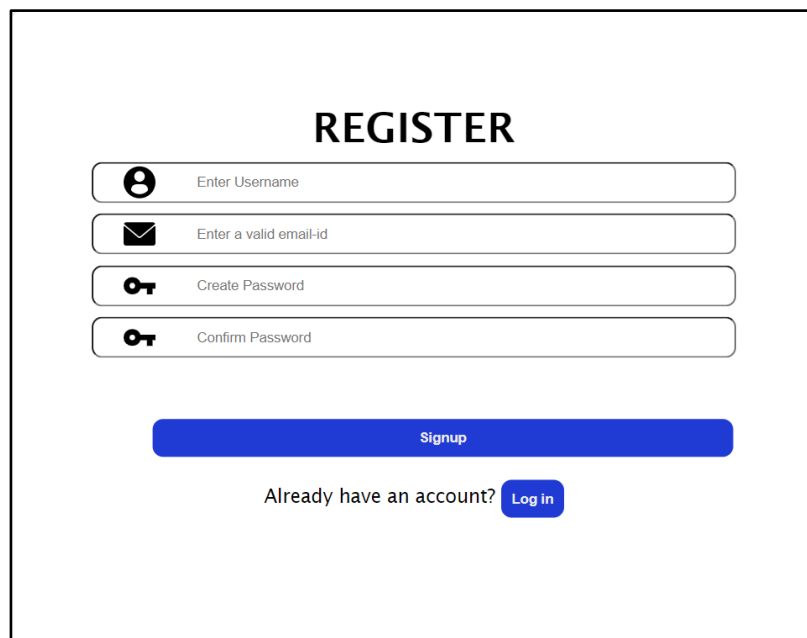
Login with username and password

[Log In](#)

or

New User? click below to Sign Up

[Sign Up](#)



**REGISTER**

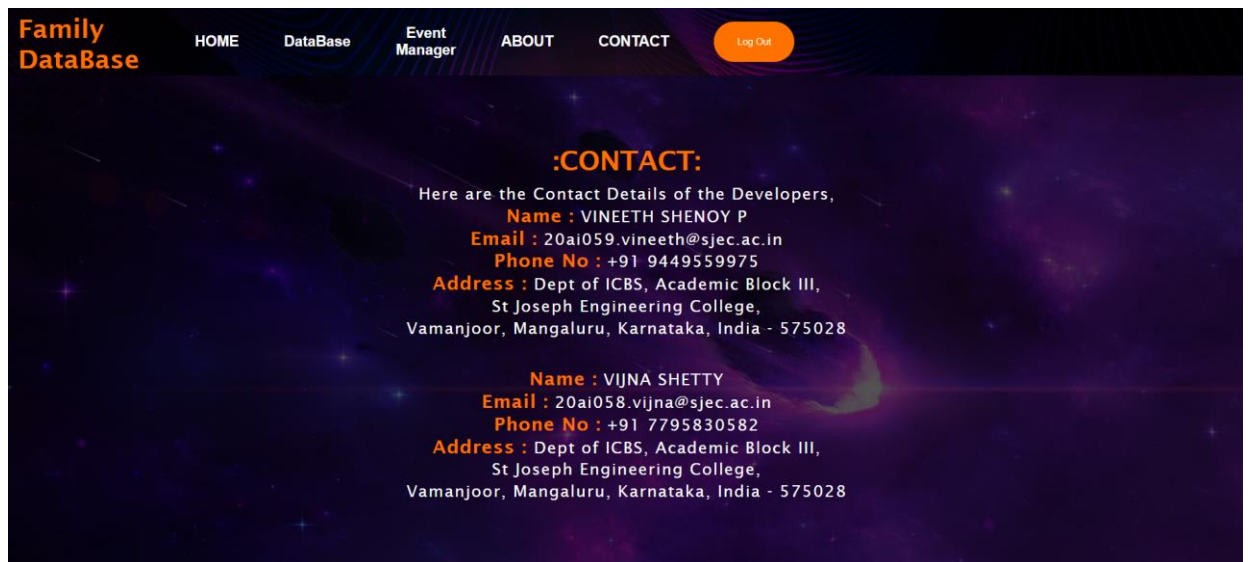
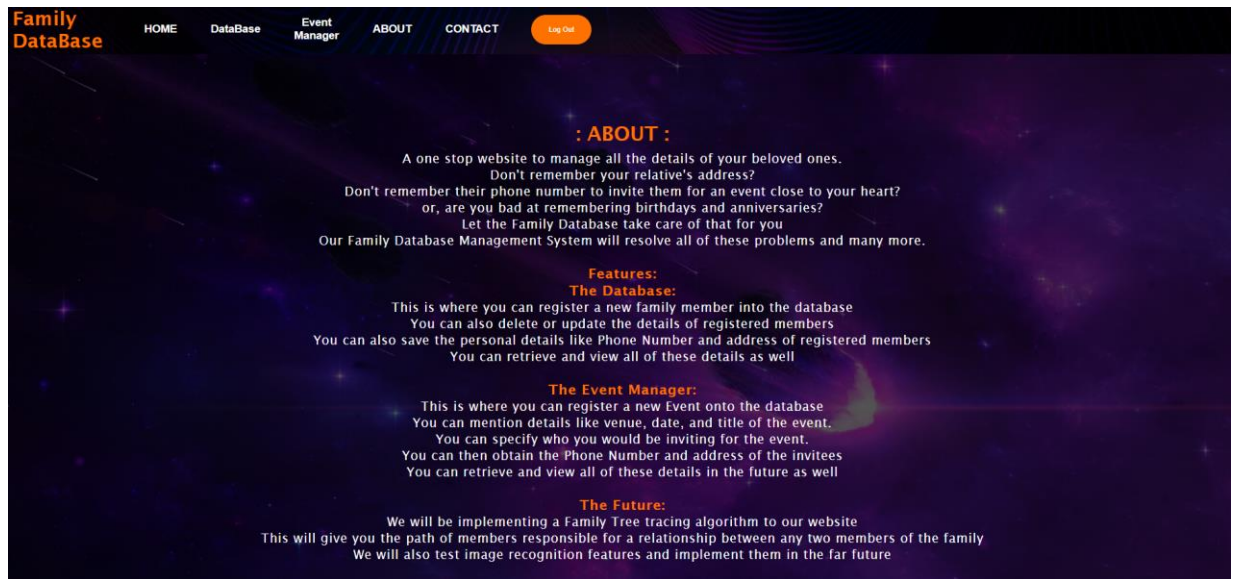
[Signup](#)

Already have an account? [Log in](#)

### 4.1.2 Home Page [3]



### 4.1.3 About and Contact



## 4.2 DATABASE FEATURES

### 4.2.1 Main Page [3]



#### 4.2.2 Insert, Update, Delete, Couple register, Parent register, Personal info register. [4]

### NEW MEMBER REGISTER

Enter First Name

Enter Name

Enter Last Name

Enter Date of Birth:

dd - mm - yyyy

Enter Gender (M-male, F-female)

SUBMIT

### UPDATE DETAILS

Enter FamMemberID (\*\*required)

UPDATE Gender (M-male, F-female) ?

Enter Date of Death if applicable:

dd - mm - yyyy

\*other details cannot be changed

SUBMIT

### DELETE MEMBER

Enter FamMemberID

SUBMIT

### REGISTER COUPLE

Enter ID of Husband

Enter ID of Wife

Enter Date of Marriage:

dd - mm - yyyy

\*Both of them must, first be registered as Family Members

SUBMIT

### REGISTER PARENTS

Enter ID of Child

Enter parent's couple ID

\*The child must be registered as a member  
\*parents must be registered as couple

SUBMIT



**PERSONAL INFO PAGE**

Search personal Details

Or Enter new info:

Enter FamMemberID

Enter valid email id

Code(Eg: +91)

Enter Phone Number

Enter your address (Exclude City, Country & Pincode)

Enter your city

Enter your country

Enter pincode

If you have a job, enter job designation  
If studying, Type "(your course) Student"  
(Eg: B.Tech Student)

Enter Job Designation/student

Enter Company/College/school Name

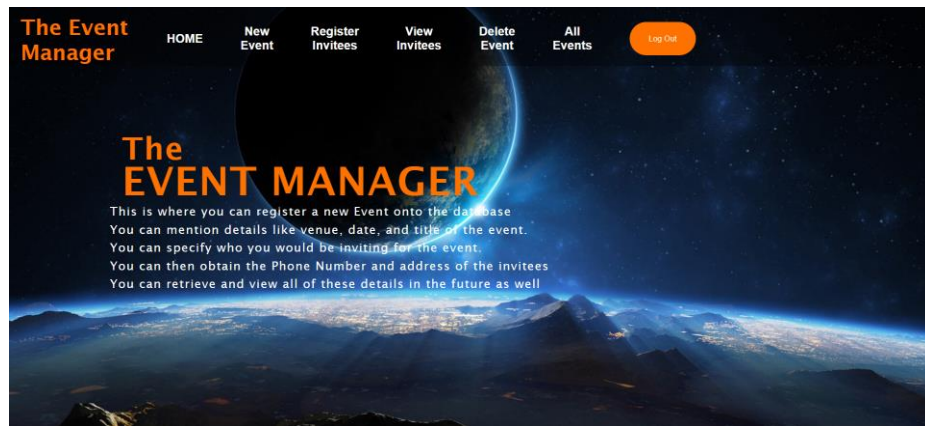
Submit

### 4.2.3 View [5]



## 4.3 EVENT MANAGER FEATURES

### 4.3.1 Main Page [3]



### 4.3.2 New Event, Delete Event [4][2]

### REGISTERING AN EVENT

Enter Name of Event

Enter Exact Venue, and other details ( Time if applicable)

Enter Date of Event:

dd-mm-yyyy

Submit

### DELETE EVENT

Enter Event\_ID

SUBMIT

### 4.3.3 Register Invitees [4][2]

### INVITEES FOR EVENT

Enter EventID

For the Invitee\_ID:  
Enter an individual's FamMemberID to invite one member only

Enter FamMemberID

OR

Enter the Couple\_ID to invite a particular couple

Enter Couple\_invited

OR

Enter the family ID to invite the entire family

Enter Family ID

Submit



## **CHAPTER 5**

# **CONCLUSION AND FUTURE WORK**

## 5.1 CONCLUSION

The web-based system for managing and inviting family members to events has been developed and implemented successfully. The system provides a user-friendly and efficient way of managing family member information and event invitations, ensuring the security of the stored data, and is easily extensible to meet various use cases. The system provides features like event management, family member management, couple management, family management, personal information management, parent/children management, event invitation, user management, security, etc.[1]

The system will be helpful for organizations and individuals who are looking for an easy way to manage and invite family members to events. The system will help them to keep track of their events, family members and personal information. The system will also be helpful for researchers who can use the data to analyse family relationships and other aspects.[1]

Overall, this project has been a valuable experience and has resulted in a functional and useful system that can be used by organizations and individuals to manage and invite family members to events.[1]

## 5.2 FUTURE WORK

One of the future updates on the system will be the ability to trace family relationships between any two members and displaying the family tree. This feature will allow users to easily view and understand the family relationships within their family. Users will be able to see the relationship between any two members and the entire family tree in an organized way. This feature will be useful for researchers who can use the data to analyse family relationships and other aspects.

Another future update will be the ability to download the invitees list of an event in an excel sheet. This feature will allow users to download the invitees list in a spreadsheet format, which can be easily shared and used for various purposes. This feature will be very

useful for organizations and individuals who are looking for an easy way to manage and invite family members to events.

In conclusion, the system is designed to be easily extensible to meet the needs of various organizations and individuals. The system is currently functional and useful, but there are still some features that can be added to improve its functionality and usability. The ability to trace family relationships and displaying the family tree and the downloadable excel sheet of invitees are just some of the many features that can be added to the system in the future.[1]

## REFERENCES

- [1] Chat GPT, by openai.com. <https://chat.openai.com/chat>.
- [2] “Creating Login, Signup Page in Django Using Mysql Database | Django Python Tutorial”, by “Progress with Python”. <https://youtu.be/tytTloigrd8>.
- [3] “Django Tutorial Part 5: Creating our home page” from [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Home\\_page](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Home_page).
- [4] “Models in Django: Saving Form Data To The Database | Django Tutorial #10”, by “ProgrammingWithHarry” <https://youtu.be/To879URvX0c>.
- [5] “Project 2: Fetching Data from our Models for ToDos List Django App”, by “ProgrammingWithHarry” <https://youtu.be/KaJBkELJ5n0>.
- [6] “How To Create Login Form In HTML and CSS | Make Sign In Form Design” by “Easy Tutorials” <https://youtu.be/OWNxUVnY3pg>.

## Other Documentations referred for Debugging and self-learning:

1. <https://www.tutorialspoint.com/django/>.
2. <https://www.webforefront.com/django/>.
3. <https://docs.djangoproject.com/>.

## Credits

Webpage Background: <https://best-wallpaper.net/> <https://4kwallpapers.com/>.